7-19-2021

Deep Learning Models to Characterize Smooth Muscle Fibers in Hematoxylin and Eosin Stained Histopathological Images of the Urinary Bladder

Sridevi Kayyur Subramanya
ss9423@rit.edu

RIT

Deep Learning Models to Characterize Smooth Muscle Fibers in Hematoxylin and Eosin Stained Histopathological Images of the Urinary Bladder

by

Sridevi Kayyur Subramanya

Submitted in partial fulfillment of the requirements for the Master of Science degree in Bioinformatics

Rochester Institute of Technology
Rochester, NY
July 19, 2021

Thesis Advisor: Dr. Feng Cui, BS, MS, Ph.D., MD
Associate Professor
Thomas H. Gosnell School of Life Sciences
College of Science
Rochester Institute of Technology, Rochester, NY

Committee members:

Dr. Rui Li, BS, MS, Ph.D.
Assistant Professor
Ph.D. Program in Computing and Information Sciences
Golisano College of Computing and Information Sciences, Rochester Institute of Technology, Rochester, NY

Dr. Hiroshi Miyamoto, M.D, Ph.D.
Professor
Department of Pathology & Laboratory Medicine
University of Rochester School of Medicine and Dentistry,
Rochester, NY

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor, Dr. Feng Cui for his keen interest, faith, and providing me tons of opportunities to explore and acquire new skills in bioinformatics. His patience, enthusiasm, scientific approach, and timely advice has helped me to a great extent in accomplishing this project. I would also like to acknowledge my thesis committee members, Dr. Hiroshi Miyamoto, for proposing this research work and explaining the medical challenges pathologists face during bladder cancer staging, and Dr. Rui Li, for his thought-provoking questions, constant encouragement, and willingness to share machine learning and deep learning knowledge. I would also like to thank Dr. Ying Wang, who worked along with Dr. Miyamoto in preparing and sharing the entire image dataset for my thesis work. Finally, I am grateful my family, most importantly my husband for their constant encouragement and support throughout my program. Without all their support, I would not have completed this work efficiently. Thank you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Muscularis propria (MP) and muscularis mucosa (MM), two types of smooth muscle fibers in the urinary bladder, are major benchmarks in staging bladder cancer to distinguish between muscle-invasive (MP invasion) and non-muscle-invasive (MM invasion) diseases. While patients with non-muscle-invasive tumor can be treated conservatively involving transurethral resection (TUR) only, more aggressive treatment options, such as removal of the entire bladder, known as radical cystectomy (RC) which may severely degrade the quality of patient's life, are often required in those with muscle-invasive tumor. Hence, given two types of image datasets, hematoxylin & eosin-stained histopathological images from RC and TUR specimens, we propose the first deep learning-based method for efficient characterization of MP. The proposed method is intended to aid the pathologists as a decision support system by facilitating accurate staging of bladder cancer. In this work, we aim to semantically segment the TUR images into MP and non-MP regions using two different approaches, patch-to-label and pixel-to-label. We evaluate four different state-of-the-art CNN-based models (VGG16, ResNet18, SqueezeNet, and MobileNetV2) and semantic segmentation-based models (U-Net, MA-Net, DeepLabv3+, and FPN) and compare their performance metrics at the pixel-level. The SqueezeNet model (mean Jaccard Index: 95.44%, mean dice coefficient: 97.66%) in patch-to-label approach and the MA-Net model (mean Jaccard Index: 96.64%, mean dice coefficient: 98.29%) in pixel-to-label approach are the best among tested models. Although pixel-to-label approach is marginally better than the patch-to-label approach based on evaluation metrics, the latter is computationally efficient using least trainable parameters.

INTRODUCTION

Urinary bladder in the human body is a muscular sac in the pelvis that stores urine. The bladder is mainly composed of four different layers [1]. Urothelium is the innermost layer of the bladder. The connective tissue underlying urothelium is the lamina propria, containing muscularis mucosa (MM), blood vessels, and fibroblasts. Thick muscle layer underneath lamina propria is the muscularis propria (MP), also known as detrusor muscle. The final layer is the serosa/adventitia that covers the bladder dome. The MM and MP are the two types of smooth muscle fibers seen in the urinary bladder. The MM is composed of several thin layers of muscle fibers, often showing discontinuous, wispy, wavy fascicles, whereas the MP consists of thick muscle bundles [2].

Bladder cancer is one of the commonly diagnosed malignancies across the world, with a total of 573,278 new cases and 212,536 new deaths in 2020 [3]. It occurs when the cells that make up the bladder grow abnormally and eventually form a tumor. According to the American Cancer Society, nearly 83,730 new cases of bladder cancer and 17,200 deaths attributable to bladder cancer are estimated to occur in 2021 in the United States [4]. Bladder cancer mostly occurs in elderly people with an average age at diagnosis being 73, with men being at higher risk than women in the ratio of 3:1 [4].

Most of bladder cancers (~90%) are urothelial carcinomas, where tumor originates in the urothelial cell lining inside of the bladder [5]. The other uncommon histological types of bladder cancers are squamous cell carcinoma (1-2% of overall bladder cancers), adenocarcinoma (about 1%), small cell carcinoma (<1%), and sarcomas (very rare) [6, 7]. Bladder cancer can be clinically divided into two distinct categories: non-muscle-invasive bladder cancer (NMIBC) (Tis, Ta, and T1) and muscle-invasive bladder cancer (MIBC) (T2-4), as shown in Figure 1. According to the Tumor, Node, Metastasis (TNM) classification [8, 9], Tis stage (carcinoma in situ) or Ta stage is

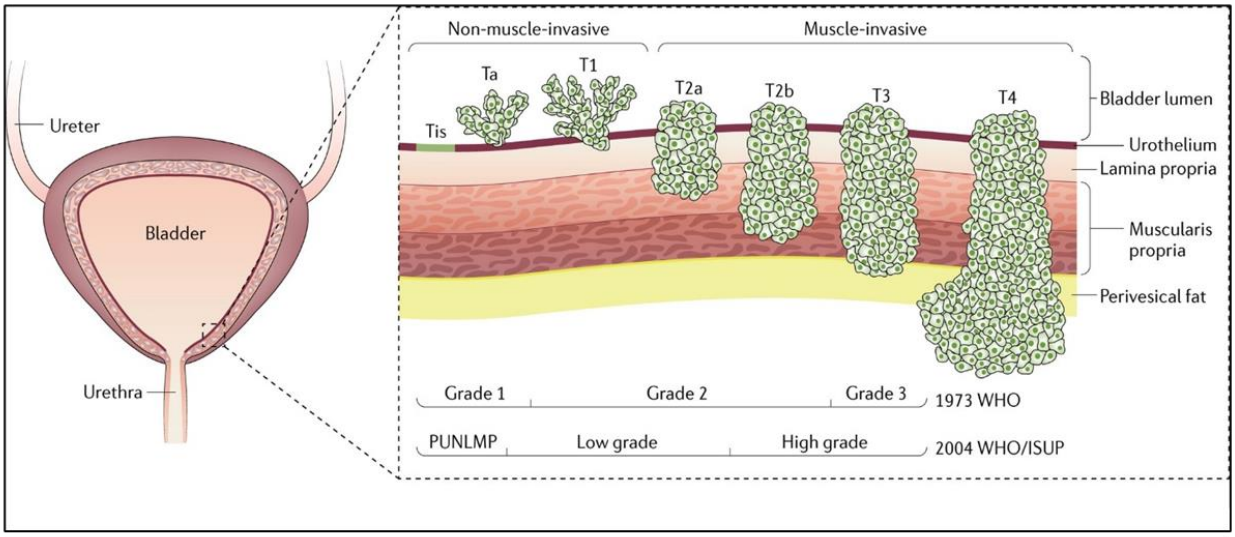

Figure 1: Pathological stages of bladder cancer [9]

a non-invasive carcinoma without or with, respectively, papillary architecture where tumor outgrows on the surface of urothelium, whereas T1 tumor invades the subepithelial connective tissue, lamina propria where MM is present. Approximately 80% of the patients are diagnosed to have NMIBC (Tis-10%, Ta-70%, and T1-20%) [10]. In MIBC, cancer invades the MP/detrusor muscle. The MM and MP tissues in the bladder are thus the major benchmarks in staging bladder cancer (T1/NMIBC vs. T2/MIBC).

Treatments for bladder cancer depend on its staging. NMIBC including cases with MM invasion can typically be managed with relatively conservative approaches like transurethral surgery, where a resectoscope is inserted into the bladder to obtain abnormal tissues for further inspection (tissues obtained are referred to as transurethral resection (TUR) or biopsy specimen), and drug therapies include instillation of BCG/mitomycin into the bladder. Whereas MIBC with MP invasion often involves aggressive treatment options like radical cystectomy (RC), where the patient's bladder is removed through surgery to avoid disease progression or metastasis (tissues obtained are referred to as RC specimen), and systemic chemotherapy. In particular, RC will have a huge impact on a patient's quality of life as there is a need for a small bag sticking permanently

around the ‘stoma’ in the abdomen to collect the urine. Thus, the distinction between MM invasion (T1/NMIBC) and MP invasion (T2/MIBC) is clinically critical.

To distinguish any component of a cell/tissue in a surgical specimen, staining mechanism is commonly adopted. Generally, all the tissue specimens obtained by TUR or RC are stained with hematoxylin & eosin (H&E), where hematoxylin stains cell nucleus blue and eosin stains cytoplasm and extracellular matrix pink. Thus, pathologists can easily differentiate the nuclear and cytoplasmic parts of the cells in an H&E-stained tissue sample. In TUR specimens exhibiting invasive cancer, it is often difficult to distinguish between the MM, which may be hyperplastic, and the MP, which may be partially destroyed or splayed by infiltrating cancer [11, 12]. In H&E-stained surgical specimens, the muscle bundles of MM, without or with cancer invasion exhibit 3 typical patterns as shown in Figure 2, a continuous layer, scattered layer showing mild hyperplasia,

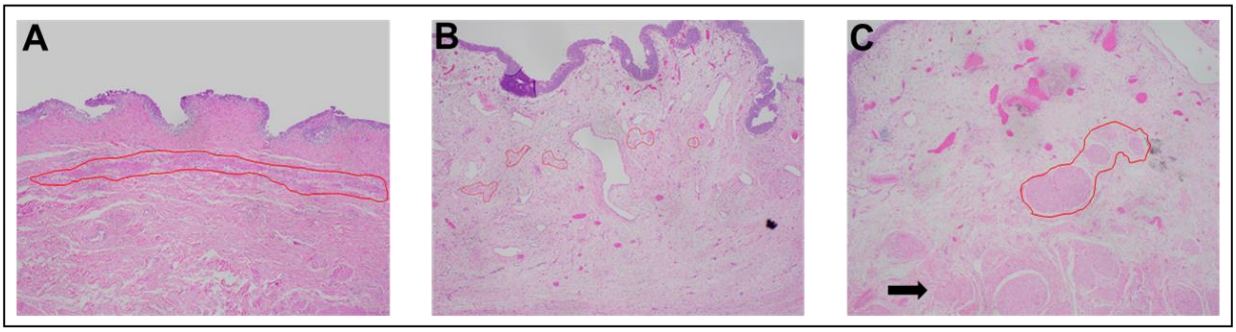

Figure 2: Muscularis mucosae (MM) muscle bundle patterns; (A) Continuous MM, (B) scattered MM, and (C) hyperplastic MM (*arrow indicates muscularis propria*)

and hyperplastic layer that mimics compact MP. Thus, MM could show no significant morphological differences to that of MP. These anomalous patterns of MM lead to misinterpretation of bladder cancer stages. Immunohistochemistry for smoothelin, a cytoskeletal protein specific stain for smooth muscle cells, has been used for differentiating MM (no or weak staining) from MP (strong staining) [13-16]. However, staining conditions have been found to considerably affect the staining intensity, and smoothelin immunohistochemistry is no longer used in the histopathological diagnosis of bladder cancer. To date, there are no other available

biomarkers that are useful for objectively distinguishing the two types of muscle bundles in bladder specimens. Therefore, it is often impossible for pathologists to differentiate MM invasion and MP invasion in biopsy specimens of bladder cancer. Thus, the distinction between MM invasion only (stage T1) and MP invasion (stage T2 or higher) is clinically critical. Particularly, considering the treatment regime for bladder cancer and challenges associated with pathological staging, detection of MP muscle fibers in H&E-stained tissues from the bladder is of high clinical importance and less visually taxing compared to MM muscle fibers. Hence, our goal in this study is to accurately differentiate MP from all non-MP tissues (including MM), using H&E-stained RC and TUR specimens.

In recent years, the use of histopathological images from H&E-stained tissue specimens to identify tissue structures/abnormalities has gained prominence mainly due to the advancements in the modern machine learning (ML) [17-21] and deep learning (DL) [22, 23] approaches, that have achieved state-of-the-art results in the field of image processing, especially for histopathological tasks such as cancer detection, tumor stage classification, and survival predictions. Generally, studies have employed two types of approaches for such tasks. In one of the approaches, the whole or some parts of the images were used to extract image-level features, and architectures such as the convolutional neural network (CNN) [24, 25] were mainly used for this purpose. Although the CNN helps us classify the whole image as a particular class, for tasks that require both prediction and localization, where we want to segregate a region inside an image to a particular class, a natural solution would be to classify each pixel of the image as a particular class. Hence, the other approach employed by different studies was to semantically segment [23, 26-28] the image into regions that belong to different classes. Most of the studies in bladder cancer have used patch-based approaches for either detection of tumor [29] or classification of the tumor stages [21, 30].

However, to our knowledge, none of the studies have evaluated ML/DL models to characterize the muscle fibers in the bladder that are critical for cancer staging.

In this work, we present a supervised DL-based framework to automatically extract informative features from H&E-stained histopathological images from RC and TUR and perform a binary classification to differentiate each pixel in those from TUR, into MP and non-MP regions. Particularly TUR specimen because there exists scope for further treatment. Since morphological differentiation of MP and non-MP is often challenging, we intend to evaluate features both at patch and pixel level. For this purpose, we introduce two approaches, patch-to-label and pixel-to-label. In the patch-to-label approach, patches are extracted such that each patch is labelled as either MP or non-MP. Prominent CNN-based architectures like VGG16 [31], ResNet [32], SqueezeNet [33], and MobileNet [34, 35] are selected for characterizing MP and non-MP regions. Whereas, in the pixel-to-label approach, every original image and the corresponding mask image are divided into an equal number of image patches and mask patches, with each pixel in the mask patch representing the label (MP or non-MP) of the corresponding pixel in the image patch. The state-of-the-art semantic segmentation models chosen for characterizing MP and non-MP regions are U-Net [36], MA-Net [37], DeepLabv3+ [38] , and FPN [39]. In both approaches, we use either patch-based inference and/or whole-image based inference to evaluate semantic segmentation-based metrics and compare the performance of all trained models. Given a TUR specimen at 100X total magnification, the produced framework is able to produce a binary and marked image representing MP and non-MP regions. Thus, we intend to ultimately be able to use the proposed work as a decision support system to highlight MP regions involved by bladder cancer in surgical specimens where pathologists are unable to do morphologically.

MATERIALS AND METHODS

The aim of the proposed framework, as illustrated in Figure 3, is to semantically segment the H&E- stained TUR specimens into MP and non-MP regions. The proposed framework consists of mainly two steps: model training and model inference. During model training, the input images were first pre-processed, and patches of defined size were extracted. These patches with the corresponding labels were passed through DL architectures to learn MP and non-MP regions in the H&E-stained tissue images. During model inference, the test images were first pre-processed and passed through the trained model to obtain the output images with predicted MP and non-MP regions. Subsequently, the output images were post-processed to obtain predicted output image.

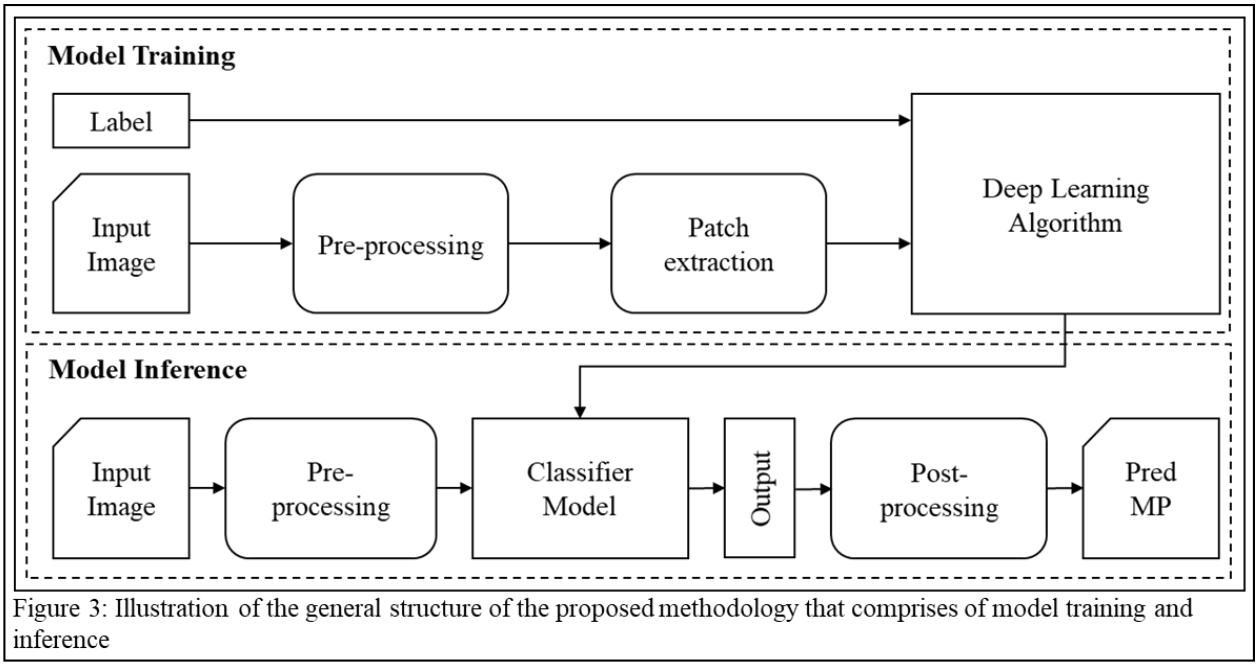

Figure 3: Illustration of the general structure of the proposed methodology that comprises of model training and inference

To semantically segment the H&E-stained TUR images into MP and non-MP regions, we applied two different approaches: 1) Patch-to-label and 2) Pixel-to-label. In the patch-to-label approach, the training patches were extracted where each patch had a single label, either MP or non-MP. Traditional CNN based architectures were used for model training and inference. In the pixel-to-label approach, training patches were extracted where each pixel of the patch had a label,

either MP or non-MP. Traditional semantic segmentation-based architectures were used for model training and inference. A detailed explanation of the proposed method is explained in the following subsections.

Histopathological Images

Upon approval from the Institutional Review Board at the University of Rochester Medical Center, a total of 303 images of H&E-stained bladder tissues from RC (237 images of size 1920 x 1440 pixels) and TUR (66 images of size 2448 x 1920 pixels) were collected from the Department of Pathology and Laboratory Medicine at University of Rochester Medical Center. The images were captured under 100X total magnification using an Olympus BX43 microscope attached with a high-resolution camera (DP27). The images were manually segmented by expert pathologists into MP and non-MP regions. Twenty-six out of 303 images were from ambiguous cases because they either contained both MM and MP tissues in the same image (RC-13 images and TUR-1 image) or it was difficult for the pathologists to morphologically distinguish between MM and MP regions (RC-10 images and TUR-2 images). Hence, 277 images were used to train and test the state-of-the-art models.

Ground Truth Preparation and Data Pre-processing

The images annotated by the pathologists were used to prepare ground truth labels. Figure 4 shows the procedure to obtain the labels for a histopathological image. A freehand drawing tool based on GrabCut segmentation algorithm [40] was used to manually mark the region of interest. The output from the tool was a bi-level mask which was then converted to a binary image. We could observe that the MP and non-MP tissues were represented as white and black regions

corresponding to pixel values of 255 and 0, respectively.

All TUR images and the corresponding binary ground truth images were resized to 1920 x

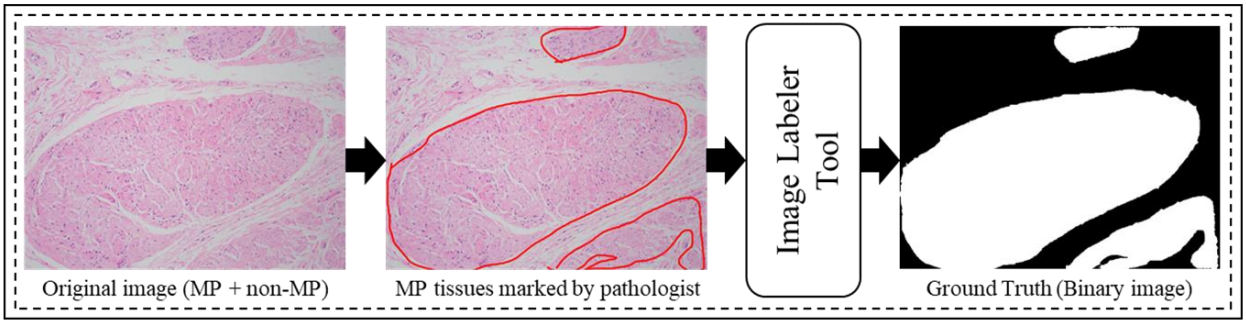

Figure 4: Illustration of generating labels from pathologists annotated images

1440 pixels using bilinear and nearest-neighbor interpolation, respectively, to maintain uniformity across all images. In the dataset, we observed considerable variability in the staining intensity among the images, especially between those from RC versus TUR. To alleviate these staining intensity differences, we applied Reinhard stain normalization [41], a standard color transferring technique that imparts the color of a chosen reference image to all the images of the dataset. This normalization resulted in a dataset with uniform stain consistency among all images, as shown in Figure 5. These stain normalized images were used as an input dataset for our analysis. Since the proposed method aimed to segment the TUR images into MP and non-MP regions, we divided our input dataset into training set consisting of 100% RC and 50% TUR images and a testing set consisting of the remaining 50% TUR images. Details of the criteria for dividing the input dataset into training and test were provided in the results section. We now provide an explanation

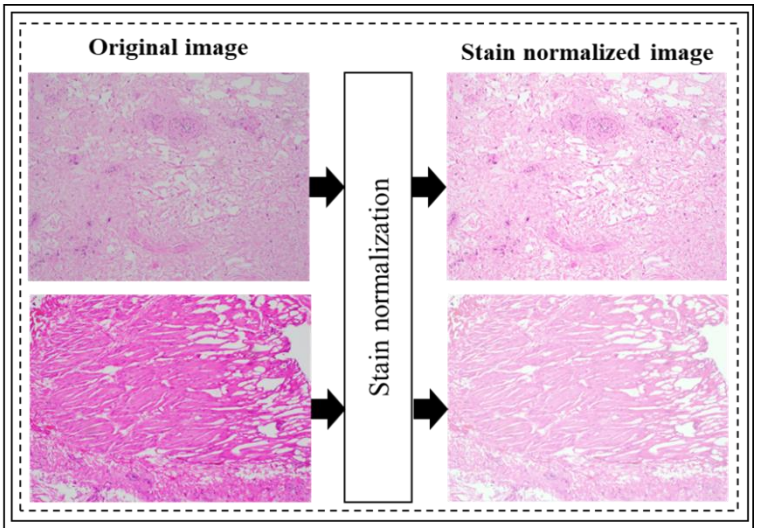

Figure 5: Representation of the effect of Reinhard stain normalization

on the two different approaches and start with the patch-to-label approach.

Patch-to-label Approach

Figure 6 shows the overview of the patch-to-label approach. The model training and patch-based inference steps are explained as follows.

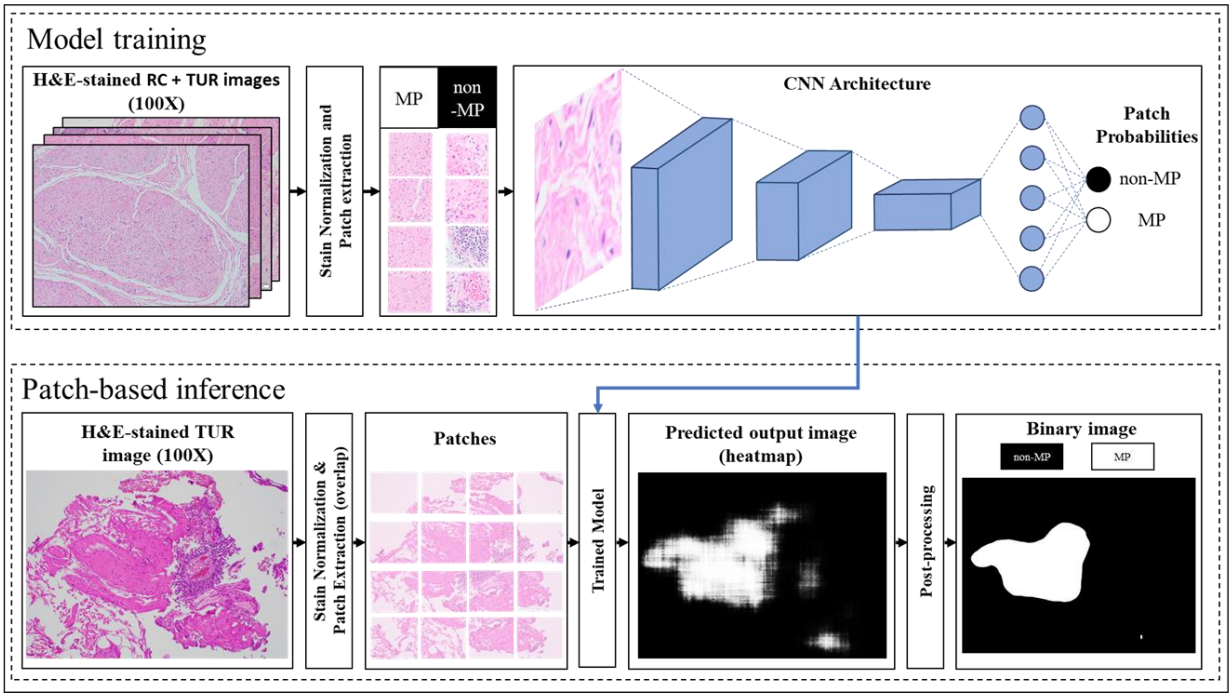

Figure 6: Patch-to-label approach for semantic segmentation of MP and non-MP regions in TUR image

In model training step, the images from training dataset were passed through stain normalization technique and were divided into several overlapping/non-overlapping patches. The patches were extracted such that each patch included either fully MP or fully non-MP regions. Thus, each patch was labelled either as an MP patch or a non-MP patch. Since each patch was a fully MP or non-MP patch, some parts of the original images were unused. Next, the dataset consisting of patches and their corresponding labels (MP v/s non-MP) was passed through different DL models that performed binary image classification. CNN is the most established algorithm for image classification among various DL models. The key advantage of CNN models is that they can self-

extract the image features through a backpropagation algorithm. The process of learning higher-level features of the image belonging to a specific class can be enhanced by increasing the depth or in other words the number of weight layers in the CNN model. The four CNN-based deep learning models chosen to characterize MP and non-MP in TUR images were VGG16, ResNet, SqueezeNet, and MobileNet. For all these architectures, instead of using random weight initialization, we chose to use the pre-trained weights from the ImageNet dataset [42]. This process of using pre-trained weights from an existing dataset is called transfer learning [43, 44], a common practice [45, 46] in ML/DL to improve image classification performance. Since the ImageNet dataset comprises of natural images and does not contain any medical-related images such as H&E-stained images, we chose to retrain the whole network and update the pre-trained weights. Also, in each of the four architectures, we changed the last layer (classifier layer) to accommodate for two-class image classification. The trained models were then used to perform patch-based inference as explained below.

In patch-based inference step, we assessed test images individually. Each image was first stain normalized and divided into overlapping patches (96% overlap). Each patch was passed through trained model that outputs the probability that a patch belonged to the MP class. The probability of each patch was assigned to the central pixel of the corresponding patch in an output image, which resulted in a small-scale heatmap representation of the predicted output image. This probability heatmap image was interpolated to the size same as that of the input image using nearest-neighbor interpolation. Each value in the heatmap representation corresponded to the probability that a fixed size patch surrounding the value was an MP region. To convert the probability heatmap to binary image representation, an optimal threshold was essential. Thus, the threshold was determined as shown in the equations below using adaptive thresholding,

$$Y_t = True\ positive\ rate - False\ positive\ rate \quad (1)$$

$$\hat{t} = \underset{t}{\text{argmax}}\, \mathbf{Y},\ \mathbf{Y} \in \mathbb{R}^{\boldsymbol{T}},\ \boldsymbol{T} = \text{number of thresholds} \quad (2)$$

Where Y_t represents the Youden's J statistic [47] which is defined as the difference between the true positive rate (Sensitivity) and false-positive rate (1 - Specificity). The true positive rate and the false positive rate were determined by comparing the probability heat map against the binary ground truth at the pixel level. A threshold $\hat{t}$ was determined such that it maximized the Youden's J statistic. The probability heatmap representation was converted to binary image representation using this threshold $\hat{t}$, where each pixel was now labelled as either MP (pixel value: 255) or non-MP (pixel value: 0). Lastly, we post-processed the binary image to remove noisy pixels and smoothen the boundaries of the binary image. Hence, we used two types of filters. First, we used a median blur filter (kernel size = 155 x 155 pixels) which reduced the noise effectively. Next, we used a simple averaging filter (kernel size = 25 x 25 pixels) to smoothen the boundary pixels. The post-processing finally resulted in a smoothed binary image representation. The same procedure was used to semantically segment all images into MP and non-MP regions in the test dataset.

To completely utilize the full image during the model training and to understand if pixel-level features better characterize the MP regions in the TUR images, we proposed the pixel-to-label approach which is explained in the following section.

Pixel-to-label Approach

Figure 7 shows the overview of the pixel-to-label approach. The model training, patch-based inference, and whole-image based inference steps are explained as follows.

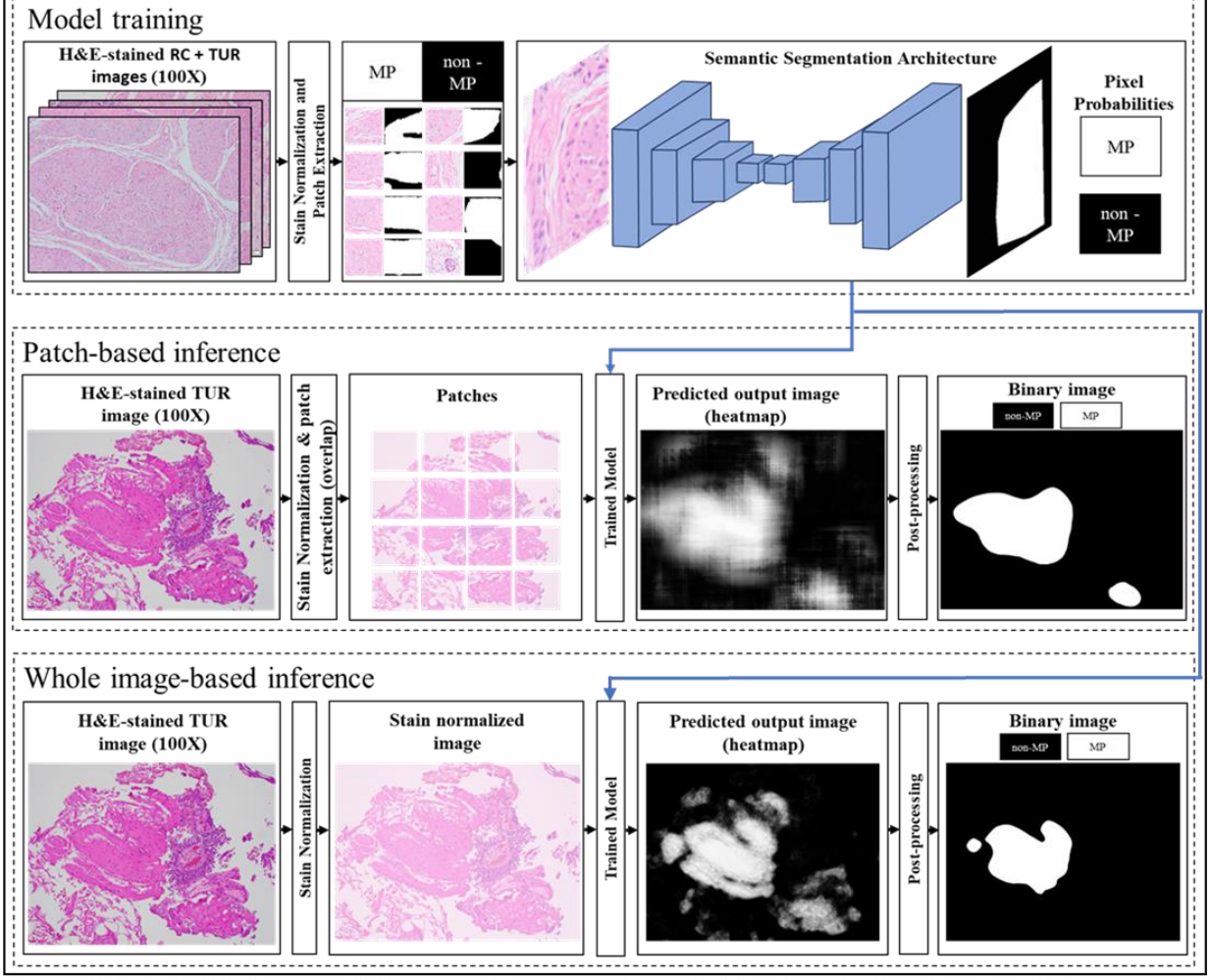

Figure 7: Pixel-to-label approach for semantic segmentation of MP and non-MP regions in TUR image

In model training step, all training images were first stain normalized. These images and their corresponding binary ground truth images were divided into overlap/non-overlapping patches. However, unlike the patch-to-label approach, the patches were extracted such that each patch included either MP or non-MP region, or both. The reason that a patch can contain any region was that each pixel of the patch was either labelled as MP or non-MP. Each extracted patch had an equal-sized relating binary ground truth mask. As a result, all informative regions of the images were effectively utilized for training the semantic segmentation models.

The input patches and the corresponding binary ground truth masks were input to the

traditional semantic segmentation-based models that performed pixel-wise binary classification. These models were developed to perform semantic segmentation tasks and their main advantage comes from the encoder-decoder architecture that forms the backbone. While the encoder encodes the original patch to a higher feature level representation, the decoder uses the same high-level feature to obtain a pixel-wise predicted output patch whose size is the same as that of the input patch and the pixel values represent the probability that a pixel is MP. The predicted output patch was compared with the ground truth mask and the weights were updated using the well-known backpropagation method. The four semantic segmentation-based deep learning models chosen to characterize MP and non-MP in TUR images were U-Net, MA-Net, DeepLabv3+, and FPN. Like patch-to-label approach, pre-trained weights from the ImageNet dataset [42] were used as initial weights before re-training the whole network and the last layer of the semantic segmentation models were changed to accommodate for pixel-wise two-class classification (MP and non-MP). The trained models were then used to perform patch-based and whole image-based inference as explained below.

To assess the trained model performance, we used two independent ways of model inference methods, patch-based and whole image-based. The working of patch-based inference method was similar to the one that was previously described in the patch-to-label approach. Here, although we obtained the MP probabilities for each pixel in a patch, we estimated an average MP probability of the patch and assigned it to the central pixel of the corresponding patch in an output image. Thus, the output image resulted in a small-scale heatmap representation of the predicted output image. This probability heatmap image was interpolated to the size same as that of the input image using nearest-neighbor interpolation to obtain predicted output image.

In whole image-based inference, the stain normalized test images were directly fed to the

trained model without dividing the images into patches. The output image was a probability heatmap image whose size was same as that of the input image and each pixel value represented the probability that a pixel was MP. Next, the optimal threshold was determined to convert the predicted probability heatmap image to a predicted output image as the value that maximized the Youden's J statistic, as shown in Equations (1) and (2). The binary image was passed through the median blur (kernel size = 155 x 155 pixels) and averaging filter (kernel size = 25 x 25 pixels) to obtain a smooth predicted binary image. The methods used in patch-based and whole image-based were applied individually to all the images of the test dataset to semantically segment each image into MP and non-MP regions.

Evaluation Metrics

As described in both patch-to-label and pixel-to-label approaches, the final output was the post-processed predicted binary image with MP and non-MP regions highlighted in different colors. To assess the performance of models used in patch-to-label and pixel-to-label approaches, we use standard pixel-level evaluation metrics like precision, recall, specificity, F1 score, mean dice coefficient, mean Jaccard index, and global pixel-wise accuracy. The basic components that describe these metrics involve true positives (TP); the total number of MP pixels in ground truth that are correctly predicted as MP, true negatives (TN); the total number of non-MP pixels in ground truth that is correctly predicted as non-MP, false positives (FP); the total number of non-MP pixels in ground truth that are wrongly predicted as MP pixels, and false negatives (FN); the total number of MP pixels in ground truth that are wrongly predicted as non-MP pixels.

The precision or the positive predicted value is the measure of correctness. It evaluates how "precisely" the model predicts the positive class, MP pixels. The precision value can be determined

as shown in equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall or sensitivity or the true positive rate corresponds to the accuracy of positive cases or in other words, MP class accuracy. It is defined as the ratio of true positives to the total number of predicted positives, as represented in equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Specificity or true negative rate determines the non-MP class accuracy. As shown in equation (5), specificity is calculated as a ratio of total true negatives to the total number of predicted negatives.

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

F1 Score is a metric defined as the harmonic mean of precision and recall, as presented in equation (6). The higher value of the F1 score signifies how well the model predicts the positive class.

$$F1\ Score = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{6}$$

Jaccard Index, also known as Intersection Over Union (IoU), is the ratio of the area of overlap between the predicted image and the ground truth image to the area of union between the predicted image and the ground truth image. We determine the mean Jaccard Index by taking an average of class specific Jaccard Indices, each for MP and non-MP using the equation (7).

$$Jaccard\ Index = \frac{TP}{TP + FP + FN} \tag{7}$$

Dice Coefficient is a statistical measure to determine the similarity between the predicted image

and the ground truth image. It emphasizes only the positive class similarity and does not account for the negative class. Thus, we determine the full image dice coefficient (MP and non-MP regions) by computing the average of class-specific dice coefficients, each for MP and non-MP using equation (8).

$$Dice\ coefficient = \frac{2 * TP}{2 * TP + FP + FN} \tag{8}$$

Mean Jaccard Index and mean Dice coefficient are the most used metrics for evaluating semantic segmentation models that show how well the model characterizes a pixel into corresponding class.

Global pixel-wise accuracy indicates the fraction of correctly predicted pixels, considering both MP and non-MP class, to the total number of pixels, and it is represented in equation (9).

$$Pixel\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

RESULTS

Software and Hardware

The proposed methodology was executed on a workstation with hardware and software specifications as described in Table 1. The workstation had 16 GB of RAM, 6 GB of graphical memory (GPU), i7 6 core processor, and Windows 10 operating system.

Hardware	Software
Random Access Memory (RAM) 16GB **Processor** Intel(R) Core(TM) i7-10750H CPU @ 2.6GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s) **Graphics** NVIDIA GeForce GTX 1660 Ti **Operating system** Windows 10 Home 64-bit (10.0, Build 19042)	**Integrated development environment** Anaconda Spyder (Python 3.8) **Libraries** PyTorch, Pandas, NumPy, OpenCV, Scikit-learn, Matplotlib **Ground truth preparation** MATLabR2020b

Table 1: Summary of machine specifications

All thesis-related tasks were performed in Spyder [48], an integrated development environment from Anaconda.org, except the Ground Truth preparation, for which MatLabR2020b Image Labeler tool was used. In Spyder, all deep learning analyses were accomplished using PyTorch [49], a Python-based scientific computing package that includes functionality to use the power of system GPUs, thereby utilizing available resources and leading to time-efficient analysis. PyTorch also incorporates automatic efficient differentiation libraries that are useful to implement deep learning neural networks. For all the numerical computations NumPy [50] library was used. For image reading/manipulations and plots, OpenCV [51] and Matplotlib [52] were utilized, respectively. For file operations such as reading the file and writing results to a file, Pandas [53]

library was used.

Deep Learning Model Architectures

We have evaluated four different models each for patch-to-label and pixel-to-label approaches, which are explained as follows.

For the patch-to-label approach, we have used state-of-the-art CNN architectures for image classification such as VGG16, ResNet18, SqueezeNet, and MobileNetV2, whose number of layers and number of parameters are listed in Table 2.

VGG16: The Visual Geometry Group (VGG) is one of the earliest CNN architecture consisting of a large number of convolutional, pooling, and fully connected layers. The model is large and has 119.55 million trainable parameters.

ResNet18: Deeper models should theoretically provide better results, however, they did not because of the vanishing gradient problem. ResNet models addressed this issue by introducing the concept of "skip connection". With this concept, we can now build deeper architectures. ResNet18 is a model which has the least depth among other models in the ResNet family with 11.18 million trainable parameters.

SqueezeNet: This model was designed to create a network with few trainable parameters without compensating for the model performance. The SqueezeNet was able to achieve the accuracy of AlexNet, however with significantly less trainable parameters as evident in the listed table (0.74 million trainable parameters).

MobileNetV2: This model is one of the models belonging to the family of MobileNet models which introduced the concept of depth-wise separable convolution that resulted in fewer parameters during training.

Patch-to-label			Pixel-to-label		
Network	**Depth (layers)**	**Trainable parameters (millions)**	**Network**	**Encoder**	**Trainable parameters (millions)**
VGG16	16	119.55	U-Net	ResNet18	14.33
ResNet18	18	11.18	MA-Net		21.68
SqueezeNet	18	0.74	DeepLabv3+		12.33
MobileNetV2	53	2.23	FPN		13.05

Table 2: Characteristics of the chosen models in patch-to-label and pixel-to-label approaches

For the pixel-to-label approach, we have used state-of-the-art semantic segmentation architectures for pixel classification such as U-Net [36], MA-Net [37], DeepLabv3+ [38], and FPN [39], whose encoder and the number of parameters are listed in Table 2. ResNet18 model was chosen as an encoder to compare the model performances of all four chosen models.

U-Net: This model was developed on top of the fully convolutional network and was used for the segmentation of tumors in the lungs and brain. The main contribution of this model was that it provides a shortcut connection between every layer in the encoder with the corresponding layer in the decoder thereby alleviating the problem of excess compression of the image happening in the encoder.

MA-Net: Multi-scale Attention Net was a model that has introduced a self-attention mechanism into the network that helps to combine local and global features. This attention mechanism provides an ability to the Manet to capture contextual dependencies.

DeepLabv3+: This model was the latest among the family of DeepLab models, which introduced the concept of atrous convolutions (dilated convolutions) and atrous spatial pyramid pooling that resulted in the enhanced model's ability in providing better segmentation results.

FPN: This model was developed to solve the problem of panoptic segmentation which was a combination of semantic and instance segmentation. The model architecture fuses two different

algorithms, ResNet152 [32] and ResNeXt152 [54] that are suited for instance and semantic segmentation of the image.

Hyperparameter Selection

To train the DL models and obtain state-of-the-art results, choosing the right model hyperparameters was important. Since we were using two different approaches (patch-to-label and pixel-to-label) to segment the TUR images into MP and non-MP regions, separate hyper-parameters were used for both approaches. Based on the combination of trial-and-error experimentation using a validation dataset and suggested standard hyperparameters settings presented in the literature, we used the model hyperparameters as listed in Table 3. The batch size

Hyper-parameters	Patch-to-label approach	Pixel-to-label approach
Batch size	32	4
Epochs	30	50
Optimization algorithm	Stochastic Gradient Descent	Adam (beta1=0.9, beta2=0.999)
Learning rate	0.001	0.0001
Criterion/ Loss function	Cross Entropy	Cross Entropy

Table 3: Hyper-parameters used for patch-to-label and pixel-to-label approaches

indicates the number of training samples processed in one iteration before updating the model weights and epoch defines the number of times the learning algorithm will encounter the entire training dataset.

For the patch-to-label approach, we used 30 epochs and we chose a batch size of 32 which is a standard batch size used in CNN models for image classification. The learning rate was chosen to be 0.001 and the stochastic gradient descent optimization algorithm was used to update the

weights. We used the cross-entropy loss function, which is generally used in image classification tasks.

For the pixel-to-label approach, since we used semantic segmentation architecture, a higher number of epochs were needed to train (50 epochs). Smaller batch sizes are commonly used for semantic segmentation tasks, and thus, we used a batch size of 4 in our analysis. The cross-entropy loss function was used determine the loss between predicted and ground truth mask. To optimize the loss function and update the weights we used a learning rate of 0.0001 with Adam optimizer.

Determination of Optimal Dataset Combination

The dataset used in this work comprises of both RC and TUR images. Although both the images are from the urinary bladder, the tissue extraction process and conditions of the tissues which were imaged are different as explained in the Introduction section. Also, our dataset comprises a higher number of RC images in comparison with the TUR images, as it is easy to label MP regions in RC than in TUR images. As a result, the reasonable question we wanted to answer was whether RC images are enough to train the model. Otherwise, should we use RC images and some TUR images for training to obtain better segmentation for TUR images? Out of curiosity, we also wanted to determine what would be the effect of just using the TUR images for training and testing, given their small number. Hence, we wanted to assess what combination of RC and TUR images was required for obtained accurate segmentation prediction for the TUR images. Therefore, we created three categories: a) Training on 100% RC and testing on 100% TUR images, b) Training on 85% TUR and testing on 15% TUR images, and c) Training on 100% RC and 50% TUR images and testing on 50% TUR images. The purpose of the testing dataset is mainly to determine the final predictions and assess all the semantic segmentation-based evaluation metrics.

One of the primary requirements for DL models is that they require a large number of training data. While dividing the whole image into patches, the use of overlapping patch extraction results in an increased number of patches for training purposes. Hence, we also wanted to confirm whether there exists any effect of using overlapped patches in comparison with non-overlapped patches. Thus, for each train/test division category, we used two types of overlapping strategies: a) No overlap and b) 50% overlap. The three train/test division categories and two overlapping strategies were tested only on the patch-to-label approach and thus resulted in a different number of training and testing patches. When we looked at the distribution of MP and non-MP classes, we found that there was a larger number of non-MP patches in comparison with MP patches as evident in the table. Hence for each combination of train/test division categories and overlapping strategies, we wanted to determine the impact of using a weighted cross-entropy loss function that provides high weight for the MP patches in comparison with non-MP patches and the weight provided is equal to the reciprocal of the number of patches belonging to a particular class. Hence, we used two weighting approaches: a) Weighted cross-entropy and b) unweighted cross-entropy. Thus, a total of 12 task combinations were assessed using the ResNet18 model in patch-to-label approach. For evaluation, we determined the mean Jaccard index, mean dice coefficient, and area under the curve (AUC) of Receiver Operating Characteristic curve (ROC)/ Precision-Recall curve (PR). Table 4 shows the mean Jaccard index and mean dice coefficient metrics for each combination at the pixel level. We observed that by using 100% RC and 50% TUR images, producing non-overlapping patches, and by using weighted cross-entropy loss function for training, the best evaluation metrics were obtained by testing the remaining 50% TUR images. The mean Jaccard index and Dice coefficient for the best test case (task no 10) was 0.95 and 0.97, respectively. For the same combination, we also observe high AUC-PR and AUC-ROC values of 0.98 and 0.99 respectively

Task	Overlap (# of patches)	Cross entropy	Mean Jaccard Index	Mean Dice Coefficient	Task no
Training: 100% RC :- 214 images Testing: 100% TUR :- 63 images	No overlap Train patches: 7,077 MP: 2,211; non-MP: 4,866	Unweighted	0.81	0.90	1
		Weighted	0.78	0.88	2
	50% overlap Train patches: 23,813 MP: 7,395; non-MP: 16,418	Unweighted	0.72	0.84	3
		Weighted	0.76	0.87	4
Only TUR images Training: 85% TUR :- 53 images Testing: 15% TUR :- 10 images	No overlap Train patches: 1,542 MP: 396; non-MP: 1,146	Unweighted	0.72	0.84	5
		Weighted	0.73	0.84	6
	50% overlap Train patches:5,029 MP: 1,357; non-MP: 3,672	Unweighted	0.72	0.84	7
		Weighted	0.72	0.84	8
Training: 100% RC + 50% TUR :- 214 + 31 = 245 images Testing: Remaining 50% TUR :- 32 images	**No overlap Train patches: 8,050 MP: 2,459; non-MP: 5,591**	Unweighted	0.94	0.96	9
		Weighted	**0.95**	**0.97**	**10**
	50% overlap Train patches:27,008 MP: 8,233; non-MP: 18,775	Unweighted	0.90	0.95	11
		Weighted	0.91	0.95	12

Table 4: 12 task combinations to decide optimal training-testing dataset, overlap, and loss function

as shown in Figure 8. As observed, in most of the cases, weighted cross entropy loss function with no overlap gave slightly better results compared to unweighted cross entropy loss function, mainly due to class imbalance in our dataset. The next best performing combination was when all RC images were used for training and all TUR images were used for testing. Here, the mean Jaccard index and Dice coefficient dropped mainly because the model had not encountered any TUR

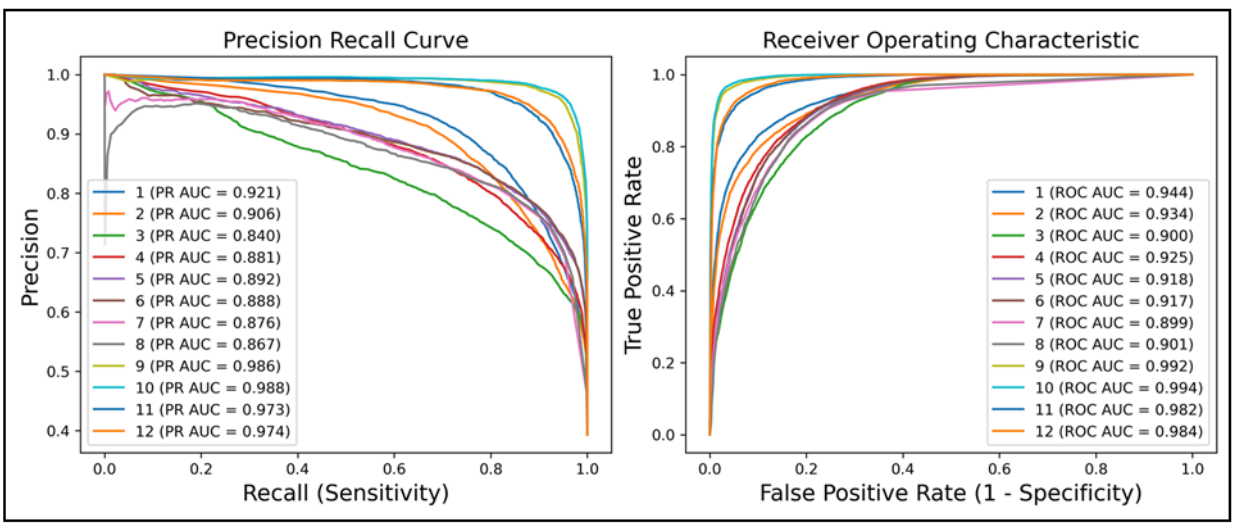

Figure 8: PR curve (left) and ROC curve (right) of ResNet18 model under 12 test cases as described in Table 4

images while training. Least performance was observed when purely TUR images (63 images) were used for both training (53 images) and testing (10 images) the model. This was expected since model was trained on minimal TUR images. Hence, for both patch-to-label and pixel-to-label approaches we used 100% RC and 50% TUR images for training purposes and used the remaining 50% TUR images for testing the trained models. We extracted non-overlapping patches and used weighted cross-entropy as a loss function for all the further analysis. Thus, a total of 214 RC images and 31 TUR images were used for training and remaining 32 TUR images were used for testing. The training images with dimensions 1920 x 1440 pixels were divided into non-overlapping patches of size 240 x 240 pixels. Two reasons for choosing a size of 240 x 240 pixels patches were that the deep learning models need the input image size to be at least 200 x 200 pixels and both the dimensions of the original image are divisible by patch size, resulting in complete utilization of input image without the need for truncation or padding. These training patches were further divided into 80% training set and 20% validation set using stratified sampling technique [55], which ensured that the distribution of the classes was maintained in both, training and validation datasets. The validation set was used as an indicator for overfitting the data during model training. Table 5 indicates the total number for training and validation patches in patch-to-label and pixel-to-label approaches.

Approach	Total patches	Class patches/pixels	Training size	Validation size
Patch-to-label	8,050	Patches - MP: 2,459; non-MP: 5,591	6,440	1,610
Pixel-to-label	11,760	Pixels - MP: 233,934,483; non-MP: 356,135,277	9,408	2,352

Table 5: Count of training and validation patches in patch-to-label and pixel-to-label approaches

Patch-to-label Model Training and Inference

In patch-to-label approach, the total number of patches extracted were 8,050 patches (Table 5). These patches comprised 2,459 MP patches and 5,591 non-MP patches, which when further divided resulted in 6,440 training patches and 1,610 validation patches. With predefined hyperparameters from Table 3, we trained all four CNN-based models. Table 6 shows the evaluation metrics for all the CNN-based models. When we compare results from four different

Evaluation metrics	VGG16	ResNet18	SqueezeNet	MobileNetV2
Precision	85.67	96.45	**96.26**	87.73
Recall	95.71	97.27	**98.12**	95.67
Specificity	89.63	97.68	**97.53**	91.33
F1 Score	90.41	96.85	**97.18**	91.53
Mean Jaccard Index	84.85	94.94	**95.44**	86.61
Mean Dice co-efficient	91.79	97.40	**97.66**	92.81
Pixelwise Accuracy	92.02	97.52	**97.76**	93.04

Table 6: Performance measure (%) of models in Patch-to-label approach

CNN-based models, we can observe that SqueezeNet performs the best. Except for precision and specificity, all the other metrics of the SqueezeNet model are higher in comparison with other models. The ResNet18 model also provides stiff competition to the SqueezeNet models. As we can observe the ResNet18 model has the high precision and specificity, and all the other metrics are very close to the best performing SqueezeNet model. The MobileNetV2 and VGG16 provide the worst results as the evaluation metrics are the least among the models. Among different metrics the mean Jaccard index and mean Dice coefficient are considered as best metrics to decide on the

superiority of the model in the segmentation tasks. We observe that the SqueezeNet/ResNet18 have mean Jaccard and Dice coefficients of 95.44/94.94 and 97.66/97.40, respectively. Whereas the MobileNetV2Net/VGG16 have mean Jaccard and Dice coefficients of 86.61/84.45 and 92.81/91.79, respectively. We also plotted the PR and ROC curves for all the models as shown in Figure 9. The AUC-PR and AUC-ROC followed a similar trend as we had seen for the evaluation

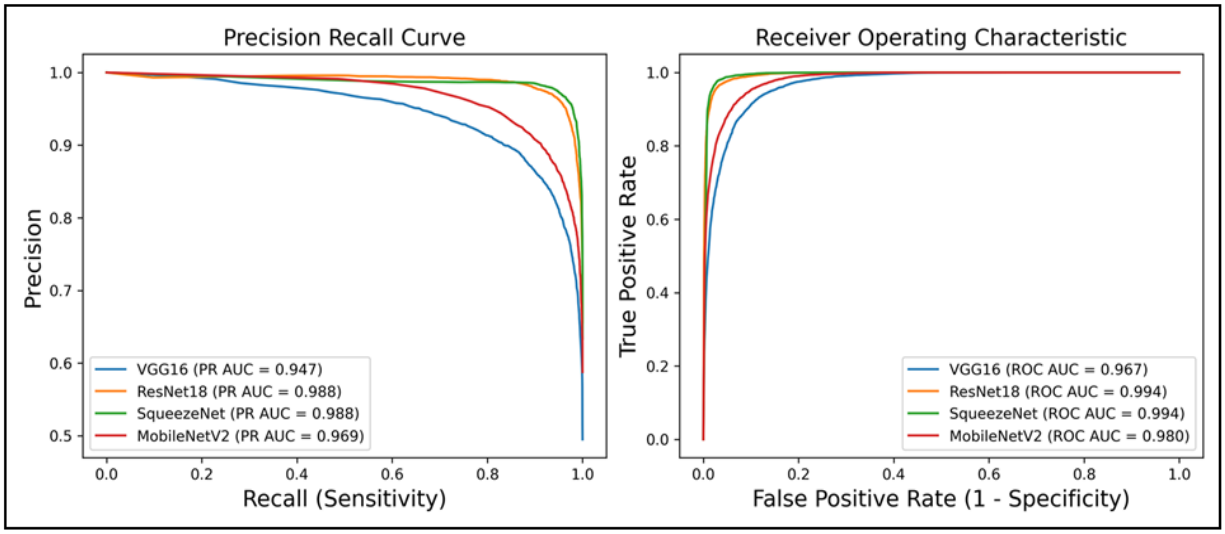

Figure 9: PR curve (left) and ROC curve (right) of all 4 CNN-based models used in patch-to-label approach

metrics. The SqueezeNet and ResNet18 models have high AUC-PR/ROC values of 0.98/0.99 in comparison with the MobileNetV2 and VGG16 models which have lower AUC-PR/ROC values. Hence, based on the evaluation metrics we conclude that the SqueezeNet and ResNet18 models are the best performing models in the patch-to-label approach.

Pixel-to-label Model Training and Inference

In pixel-to-label approach, since each pixel of the patch was labelled and for reasons mentioned in the Methods section, the total number of patches extracted were 11,760 patches which were higher than patches extracted for the patch-to-label approach (Table 5). These patches comprised 233.9 million MP pixels and 356.1 million non-MP pixels. The 11,760 patches when

further divided, resulted in 9,408 training patches and 2,352 validation patches. We used the weighted cross-entropy loss function with weighted random subsampling to ensure that each batch of size 4 saw a proportional number of MP and non-MP classes. The weight provided is equal to the reciprocal of the number of pixels belonging to a particular class. With predefined hyperparameters from Table 3, we trained all four semantic segmentation-based models. Model evaluation for patch-based and whole image-based inference types for the pixel-to-label approach is explained below.

Patch-based inference

Patch-based inference is an approach similar to that of the inference for the patch-to-label approach. Table 7 shows the evaluation metrics for all the semantic segmentation-based models.

Evaluation metrics	U-Net	MA-Net	DeepLabv3+	FPN
Precision	95.13	93.76	**97.92**	86.98
Recall	98.21	96.71	**97.56**	95.14
Specificity	96.74	95.83	**98.66**	90.78
F1 Score	96.64	95.21	**97.74**	90.88
Mean Jaccard Index	94.57	92.35	**96.35**	85.64
Mean Dice co-efficient	97.20	96.02	**98.14**	92.25
Pixelwise Accuracy	97.32	96.18	**98.22**	92.49

Table 7: Patch-based performance measure (%) of models in pixel-to-label approach

When we compare results of four different semantic segmentation-based models, we observe that the DeepLabv3+ performs the best. The DeepLabv3+ model outperforms all the other models

across all different evaluation metrics (all metrics > 95%) as highlighted in the table. The next best performing model is the U-Net, where all metrics except the mean Jaccard index are greater than 95%. U-Net is followed by MA-Net. The FPN provides the worst results as the evaluation metrics are the least among the models. We observe that the DeepLabv3+/U-Net have a mean Jaccard index and mean dice coefficient of 96.35%/94.57% and 98.14%/97.20%, respectively. Whereas the MA-Net/FPN have a mean Jaccard index and mean dice coefficient of 92.35%/85.64% and 96.02%/92.25%, respectively. We also plotted the PR and ROC curves for all the models as shown in Figure 10 (dashed lines). The AUC-PR and AUC-ROC followed a similar trend as we had seen for the evaluation metrics. The DeepLabv3+ model shows high AUC-PR/ROC values of 0.99/0.99

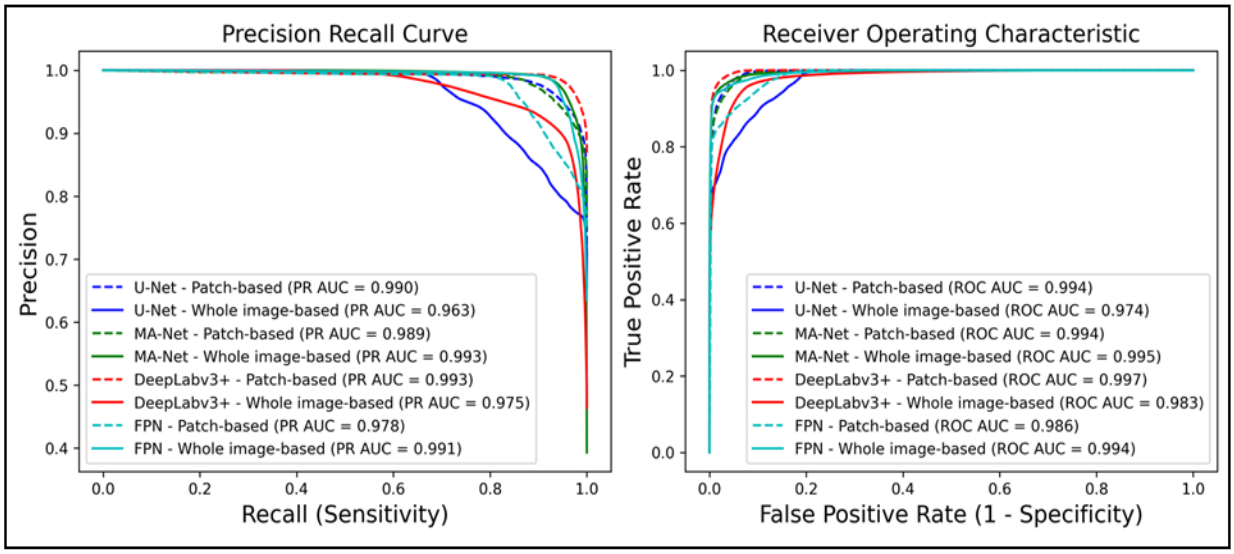

Figure 10: PR curve (left) and ROC curve (right) of all 4 semantic segmentation-based models (both patch-based inference and whole image-based inference) used in pixel-to-label approach

in comparison with other models. DeepLabv3+ is followed by U-Net and finally, FPN had the least AUC-PR/ROC values when compared to other models. Hence, based on the evaluation metrics we conclude that for patch-based inference, the DeepLabv3+ and U-Net models are the best performing models in the pixel-to-label approach.

Whole image-based inference

In whole image-based inference, the test images were not divided into patches. Instead, the full image was passed through the trained model and the predictions were achieved. Table 8 shows the evaluation metrics for all the semantic segmentation-based models using whole image-based

Evaluation metrics	U-Net	MA-Net	DeepLabv3+	FPN
Precision	85.51	**99.35**	91.30	98.85
Recall	92.45	**96.50**	96.50	94.84
Specificity	89.85	**99.59**	94.05	99.28
F1 Score	88.84	**97.90**	93.83	96.80
Mean Jaccard Index	82.79	**96.64**	90.17	94.93
Mean Dice co-efficient	90.56	**98.29**	94.82	97.40
Pixelwise Accuracy	90.87	**98.38**	95.01	97.53

Table 8: Whole image-based performance measure (%) of models in pixel-to-label approach

inference. When we compare the results of four different semantic segmentation-based models, we observe that the MA-Net performs the best. The MA-Net model outperforms all the other models across all different evaluation metrics (all metrics > 95%) as highlighted in the table. The next best performing model is the FPN, where all metrics except the recall and mean Jaccard index are greater than 95%. FPN is followed by DeepLabv3+. The U-Net is providing the worst results as their evaluation metrics are least compared to all other models. We observe that the MA-Net/FPN have a mean Jaccard and mean dice coefficient of 96.64%/94.93% and 98.29%/97.40%, respectively. The DeepLabv3+/U-Net have a mean Jaccard and mean dice coefficient of 90.17%/82.79% and 94.82%/90.56%, respectively. We also plotted the PR and ROC curves for all

the models as shown in Figure. 10 (solid lines). The AUC-PR and AUC-ROC followed a similar trend as we had seen for the evaluation metrics. The MA-Net model resulted in high AUC-PR/ROC values of 0.99/0.99 in comparison with other models. MA-Net is followed by DeepLabv3+ and finally, U-Net has the least AUC-PR/ROC values as observed in the figure. Hence, based on the evaluation metrics, we conclude that for whole image-based inference, the MA-Net and DeepLabv3+ models are the best performing models in the pixel-to-label approach.

It is very interesting to note that when we compare the two inference methods in the pixel-to-label approach, we observe that, for patch-based inference method that uses the extracted patches for image predictions, DeepLabv3+/U-Net provided the best results in comparison with MA-Net/FPN. Whereas for whole image-based inference that uses full-sized images for their prediction, MA-Net/FPN were the best performing models in comparison with DeepLabv3+/U-Net. The reason for such behavior can be attributed to the architectural structure of these models. The MA-Net [37] which stands for multi-scale attention network and FPN [39] which stands for feature pyramidal network are both built on the principle of extracting information from the images at multiple scales. When we input an image to these networks, they try to understand the contextual information by extracting the features from an image and its corresponding scaled version. So, these networks have the inherent ability to capture contextual information while predicting the class of each pixel in an image. Hence, we speculate that, when we provide a full image to these networks, the presence of full information helps them predict more accurately in comparison with their performance when we give these networks just a patch or a small part of the image. Whereas, in patch-based inference, the DeepLabv3+ [38] and U-Net [36] performed the best since the procedure explicitly considers the neighboring pixel's MP probability in a patch before assigning

the MP probability (average of all pixel's MP probability) value to its central pixel. Thus, these results show the importance of semantic segmentation model architecture in determining the region of interest in an input image.

Comparison of Patch-to-label and Pixel-to-label Approach

When we compare the results of patch-to-label and pixel-to-label approaches, based on the Jaccard index and dice coefficient metrics, we can conclude that the pixel-to-label approach is marginally better than the patch-to-label approach [56]. We anticipated this result because in the pixel-to-label approach we are using the models that are tailor-made for the semantic segmentation task and they directly coincide with our aim of semantically segmenting the TUR images into MP and non-MP regions. However, the patch-to-label approach also provides stiff competition to the pixel-to-label approach. Although pixel-to-label has a slightly better Jaccard index and dice coefficient, the time taken to train the pixel-to-label approach architecture is much higher in comparison with patch-to-label approach models. Also, the number of parameters to train for pixel-to-label models is significantly higher than the patch-to-label models except for the VGG16 model. The conclusion is that for the task of semantically segmenting the TUR images, both patch-to-label and pixel-to-label approaches work extremely well.

Visualization of Segmentation Results

We have provided a visualization of the segmentation result for test TUR images obtained from both approaches. Figures 11, 12, and 13 represent the visualization results for the patch-to-label, pixel-to-label - patch-based, and pixel-to-label - whole image-based approaches, respectively. In these figures, each row represents an original TUR image with a pathologist

marked MP region (red mark) in the first column. The subsequent columns represent the corresponding predictions from different patch-to-label approach models and pixel-to-label approach models. In Figure 11 we can observe that among all models, the SqueezeNet prediction-based visualization is the best. Similarly, in Figures 12 and 13, we can observe that among all models, DeepLabv3+ for patch-based and MA-Net for whole image-based inference provides the best visualization result in pixel-to-label approach.

We also had another test dataset for which pathologists were skeptical in deciding MP regions due to the reasons described in the Methods section. We also passed these images through the best models in both the approaches, i.e., SqueezeNet for patch-to-label, DeepLabv3+ for pixel-to-label (patch-based) and MA-Net for pixel-to-label (whole image-based), and the visualizations are represented in Figure 14.

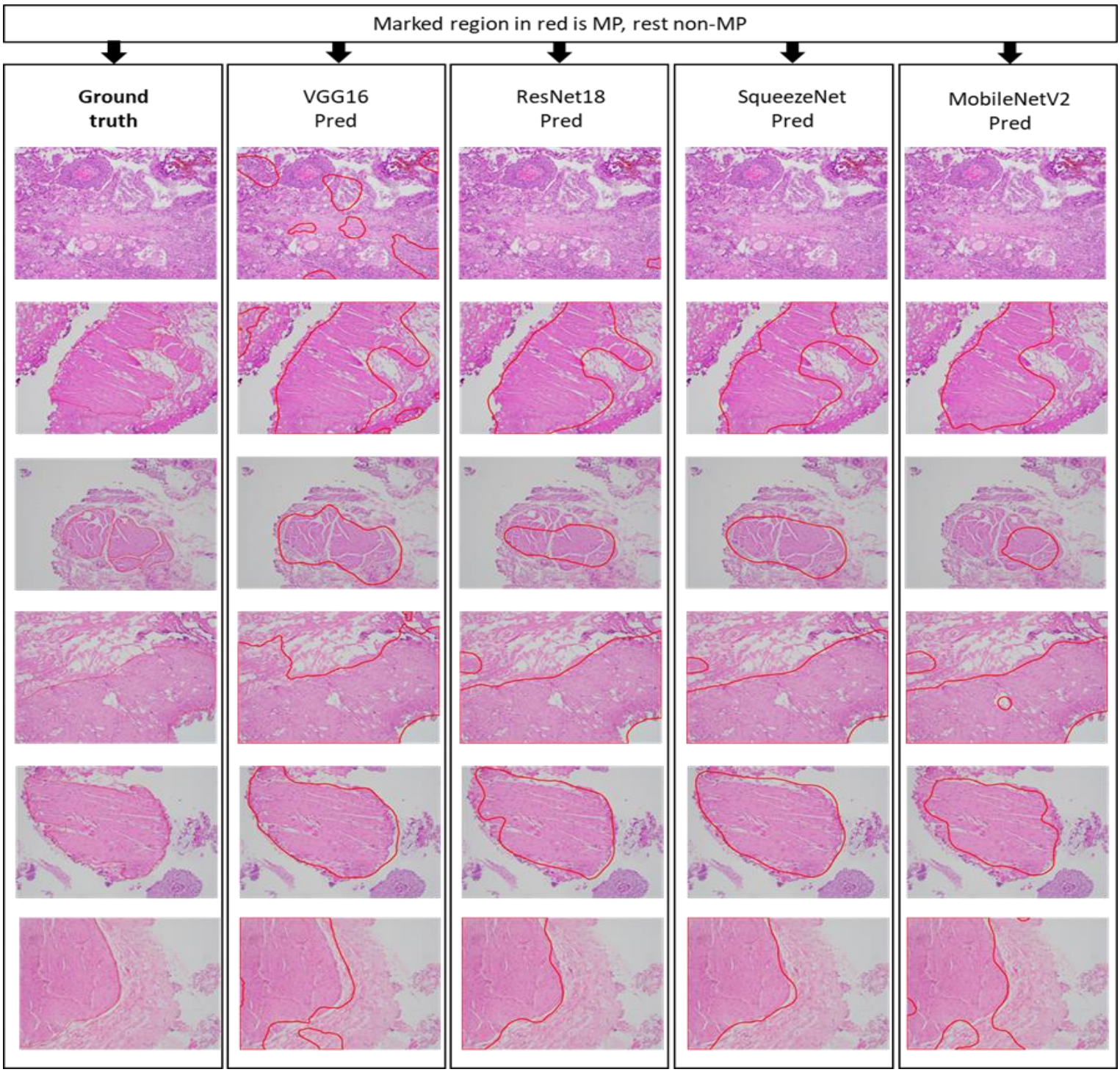

Figure 11: Visualization of segmentation results for test TUR images using trained models from patch-to-label approach

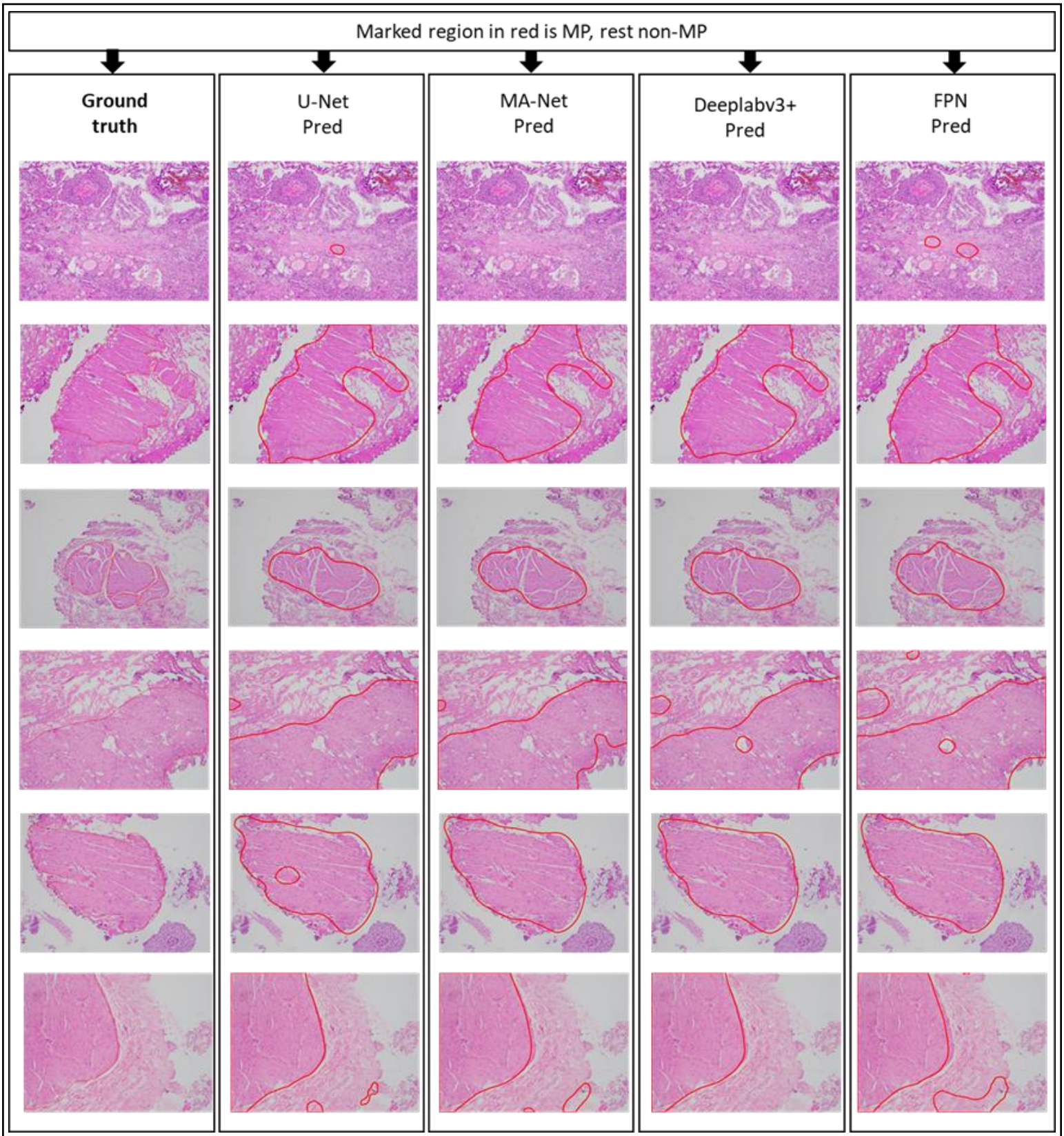

Figure 12: Visualization of patch-based segmentation results for test TUR images using trained models from pixel-to-label approach

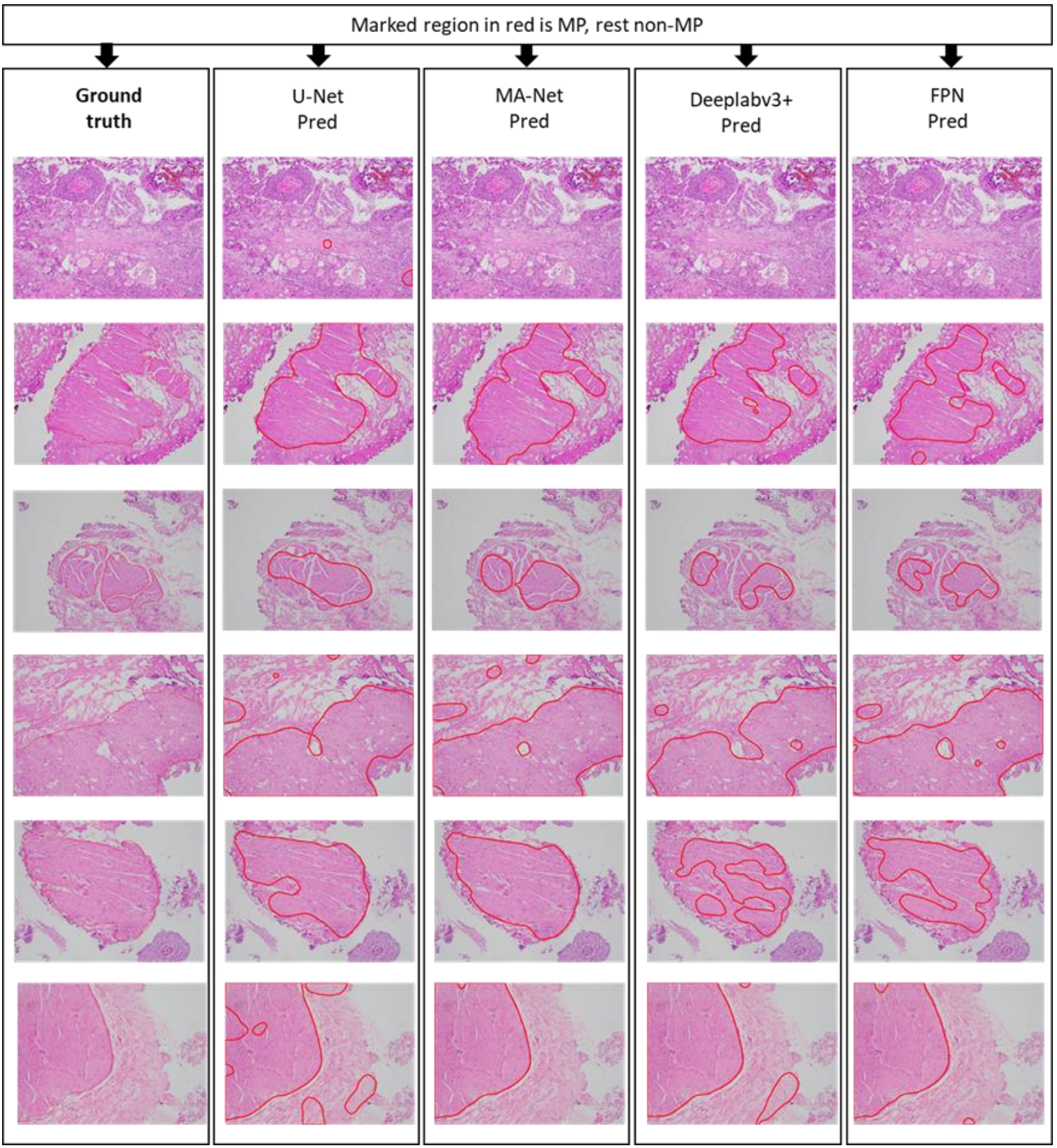

Figure 13: Visualization of whole image-based segmentation results for test TUR images using trained models from pixel-to-label approach

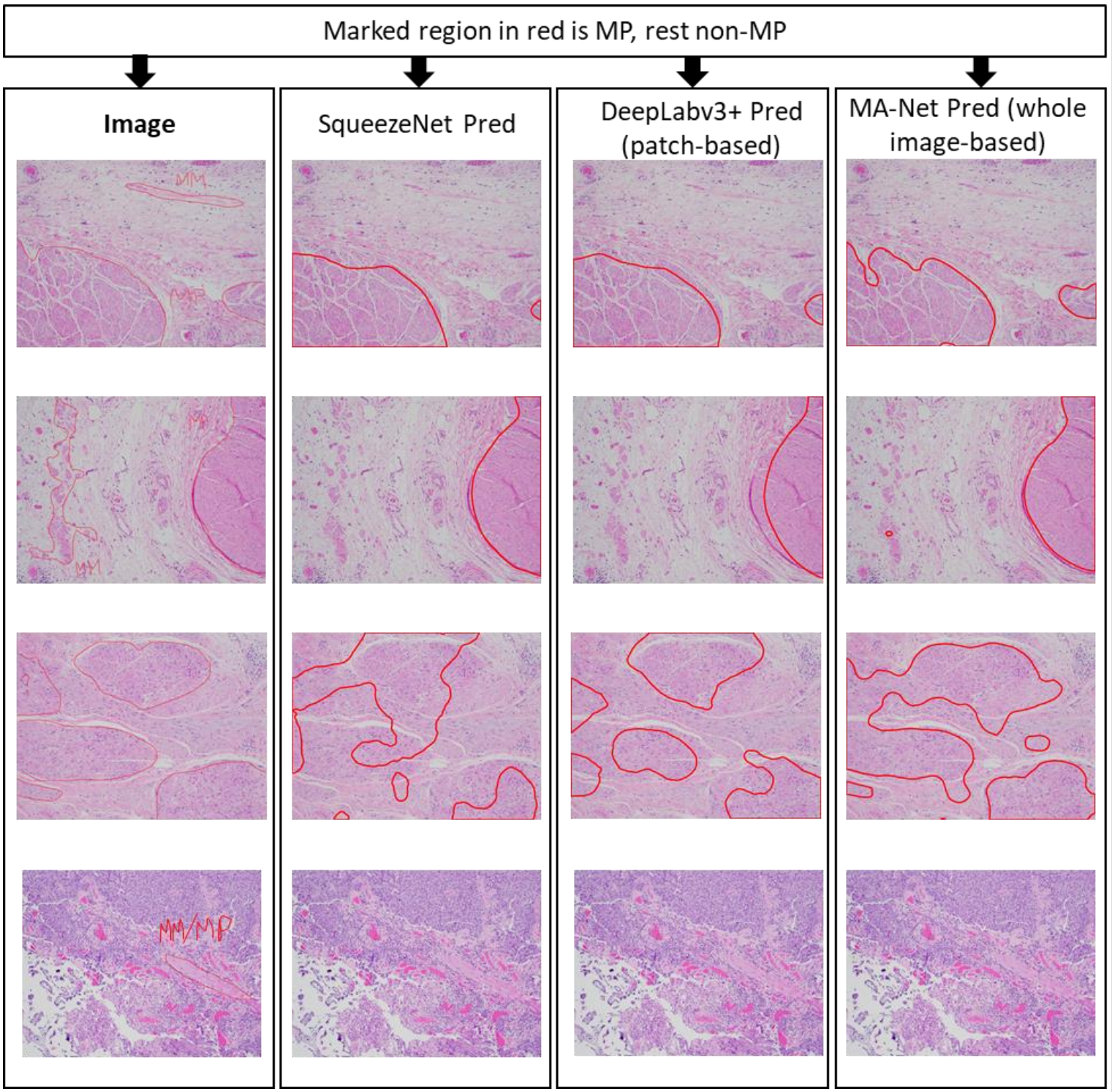

Figure 14: Visualization of segmentation results for ambiguous H&E-stained urinary bladder images using trained SqueezeNet from patch-to-label, DeepLabv3+ from pixel-to-label (patch-based inference), and MA-Net from pixel-to-label (whole image-based inference)

CONCLUSION

As MM and MP are major benchmarks in staging bladder cancer and their distinction being clinically critical, our work aimed at characterizing MP (in particular) in H&E-stained tissues obtained by TUR using DL approaches. In this study, we proposed two DL model training approaches: patch-to-label and pixel-to-label. For these, we chose 4 different state-of-the-art CNN-based architectures and semantic segmentation-based architectures and compared their performances at a pixel level. Our analysis indicated that the pixel-to-label approach was marginally better than the patch-to-label approach. MA-Net model from the pixel-to-label approach – whole image-based inference outperformed all other models, with mean Jaccard index, mean dice coefficient, and pixel-wise accuracy equal to 96.64%, 98.29%, and 98.38%, respectively. However, the patch-to-label approach used CNN-based models with reduced trainable parameters and used much less time for model training and inference in contrast with the pixel-to-label approach. Particularly, SqueezeNet, the model with the least trainable parameters (0.74 million), resulted in comparable model performance with mean Jaccard index, mean dice coefficient, and pixel-wise accuracy equal to 95.44%, 97.66%, and 97.76%, respectively. Hence, using both approaches, we were able to successfully characterize MP and non-MP regions in H&E-stained TUR specimens. It is expected that our framework will make an important contribution by acting as a decision support system to distinguish between the presence and absence of MP invasion (T2 disease) in bladder cancer specimens.

FUTURE WORK

- In this work, we have used only 31 TUR images for training along with RC images and tested the remaining 32 TUR specimens. With increased TUR specimens in both training and testing datasets, the versatility of the DL model is enhanced to determine MP and non-MP regions in complex TUR specimens.
- To explore and evaluate traditional ML models like logistic regression, decision trees, support vector machines, etc, MP and non-MP features can be extracted from the final layer of the DL models and these features can be used to train and assess the performance of the ML models.
- To explore other CNN-based and semantic segmentation-based architectures using the proposed approaches.
- An additional semantic segmentation-based technique can be implemented, where instead of using pre-trained model weights from the ImageNet dataset, we may utilize pre-trained model weights from our proposed work (patch-based model training) and input full-sized images for model training. This way, the model is purely trained on H&E-stained histopathological images. This approach requires high-performance computers and is time-consuming. However, this approach might result in improved model performance.
- To characterize MM in addition to MP in H&E-stained biopsy specimens from the bladder, thus giving rise to decision support system to accurately distinguish between MM invasion (T1 disease) and MP invasion (T2 disease) in bladder cancer specimens.
- To build a cloud-based graphical user interface to make this work more accessible to all pathologists to semantically segment MP regions in H&E-stained TUR specimens, thereby facilitating accurate staging of bladder cancer, MIBC (MP invasion) v/s NMIBC (MM/non-MM invasion).

REFERENCES

[1] S. R. Bolla *et al.*, "Histology, Bladder," *StatPearls [Internet]*, 2021.

[2] J. Dixon *et al.*, "Histology and fine structure of the muscularis mucosae of the human urinary bladder," *Journal of anatomy*, vol. 136, no. Pt 2, p. 265, 1983.

[3] H. Sung *et al.*, "Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209-249, 2021.

[4] T. A. C. S. m. a. e. c. team. "Key Statistics for Bladder Cancer." https://www.cancer.org/cancer/bladder-cancer/about/key-statistics.html#references (accessed May 29th, 2021).

[5] A. M. Kamat *et al.*, "Bladder cancer," *The Lancet*, vol. 388, no. 10061, pp. 2796-2810, 2016.

[6] T. A. C. S. m. a. e. c. team. "What Is Bladder Cancer?" American Cancer Society medical information. https://www.cancer.org/cancer/bladder-cancer/about/what-is-bladder-cancer.html#written_by (accessed 2020).

[7] M. Tretiakova. "WHO classification." http://www.pathologyoutlines.com/topic/bladderwhoisup.html (accessed November 9th, 2020).

[8] L. H. Sobin *et al.*, *TNM classification of malignant tumours*. John Wiley & Sons, 2011.

[9] O. Sanli *et al.*, "Bladder cancer," *Nature reviews Disease primers*, vol. 3, no. 1, pp. 1-19, 2017.

[10] M. S. Soloway, "It is time to abandon the "superficial" in bladder cancer," (in eng), *Eur Urol*, vol. 52, no. 6, pp. 1564-5, Dec 2007, doi: 10.1016/j.eururo.2007.07.011.

[11] G. P. Paner *et al.*, "Further characterization of the muscle layers and lamina propria of the urinary bladder by systematic histologic mapping: implications for pathologic staging of invasive urothelial carcinoma," (in eng), *Am J Surg Pathol,* vol. 31, no. 9, pp. 1420-9, Sep 2007, doi: 10.1097/PAS.0b013e3180588283.

[12] H. Miyamoto *et al.*, "Transurethral resection specimens of the bladder: outcome of invasive urothelial cancer involving muscle bundles indeterminate between muscularis mucosae and muscularis propria," (in eng), *Urology,* vol. 76, no. 3, pp. 600-2, Sep 2010, doi: 10.1016/j.urology.2009.12.080.

[13] F. T. van der Loop *et al.*, "Smoothelin, a novel cytoskeletal protein specific for smooth muscle cells," (in eng), *J Cell Biol,* vol. 134, no. 2, pp. 401-11, Jul 1996, doi: 10.1083/jcb.134.2.401.

[14] G. P. Paner *et al.*, "Diagnostic utility of antibody to smoothelin in the distinction of muscularis propria from muscularis mucosae of the urinary bladder: a potential ancillary tool in the pathologic staging of invasive urothelial carcinoma," (in eng), *Am J Surg Pathol,* vol. 33, no. 1, pp. 91-8, Jan 2009, doi: 10.1097/PAS.0b013e3181804727.

[15] N. Elkady *et al.*, "Diagnostic value of smoothelin and vimentin in differentiating muscularis propria from muscularis mucosa of bladder carcinoma," (in eng), *Int J Biol Markers,* vol. 32, no. 3, pp. e305-e312, Jul 2017, doi: 10.5301/jbm.5000252.

[16] H. Miyamoto *et al.*, "Pitfalls in the use of smoothelin to identify muscularis propria invasion by urothelial carcinoma," (in eng), *Am J Surg Pathol,* vol. 34, no. 3, pp. 418-22, Mar 2010, doi: 10.1097/PAS.0b013e3181ce5066.

[17] D. Komura *et al.*, "Machine Learning Methods for Histopathological Image Analysis," (in eng), *Comput Struct Biotechnol J,* vol. 16, pp. 34-42, 2018, doi: 10.1016/j.csbj.2018.01.001.

[18] A. El-Baz *et al.*, "Machine Learning Applications in Medical Image Analysis," (in eng), *Comput Math Methods Med,* vol. 2017, p. 2361061, 2017 2017, doi: 10.1155/2017/2361061.

[19] Y. C. Zhang *et al.*, "Machine Learning Interface for Medical Image Analysis," (in eng), *J Digit Imaging,* vol. 30, no. 5, pp. 615-621, Oct 2017, doi: 10.1007/s10278-016-9910-0.

[20] H. Asri *et al.*, "Using machine learning algorithms for breast cancer risk prediction and diagnosis," *Procedia Computer Science,* vol. 83, pp. 1064-1069, 2016.

[21] P.-N. Yin *et al.*, "Histopathological distinction of non-invasive and invasive bladder cancers using machine learning approaches," *BMC Medical Informatics and Decision Making,* vol. 20, no. 1, pp. 1-11, 2020.

[22] L.-C. Chen *et al.*, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062,* 2014.

[23] G. Litjens *et al.*, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," (in eng), *Sci Rep,* vol. 6, p. 26286, 05 2016, doi: 10.1038/srep26286.

[24] B. Wiestler *et al.*, "Deep learning for medical image analysis: a brief introduction," (in eng), *Neurooncol Adv,* vol. 2, no. Suppl 4, pp. iv35-iv41, Dec 2020, doi: 10.1093/noajnl/vdaa092.

[25] J. Gao *et al.*, "Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: An overview," (in eng), *Math Biosci Eng,* vol. 16, no. 6, pp. 6536-6561, 07 2019, doi: 10.3934/mbe.2019326.

[26] T. B. Sekou *et al.*, "From Patch to Image Segmentation using Fully Convolutional Networks--Application to Retinal Images," *arXiv preprint arXiv:1904.03892,* 2019.

[27] W. Bulten *et al.*, "Epithelium segmentation using deep learning in H&E-stained prostate specimens with immunohistochemistry as reference standard," *Scientific reports,* vol. 9, no. 1, pp. 1-10, 2019.

[28] S. A. Taghanaki *et al.*, "Deep semantic segmentation of natural and medical images: a review," *Artificial Intelligence Review,* vol. 54, no. 1, pp. 137-178, 2021.

[29] S. A. Harmon *et al.*, "Multiresolution application of artificial intelligence in digital pathology for prediction of positive lymph nodes from primary tumors in bladder cancer," *JCO clinical cancer informatics,* vol. 4, pp. 367-382, 2020.

[30] Q. Song *et al.*, "A Machine Learning Approach for Long-Term Prognosis of Bladder Cancer based on Clinical and Molecular Features," (in eng), *AMIA Jt Summits Transl Sci Proc,* vol. 2020, pp. 607-616, 2020.

[31] K. Simonyan *et al.*, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[32] K. He *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[33] F. N. Iandola *et al.*, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," *arXiv preprint arXiv:1602.07360,* 2016.

[34] M. Sandler *et al.*, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.

[35] A. Howard *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314-1324.

[36] O. Ronneberger *et al.*, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015: Springer, pp. 234-241.

[37] T. Fan *et al.*, "MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation," *IEEE Access*, vol. 8, pp. 179656-179665, 2020.

[38] L.-C. Chen *et al.*, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801-818.

[39] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[40] C. Rother *et al.*, "" GrabCut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314, 2004.

[41] E. Reinhard *et al.*, *Color Transfer between Images* (EEE Comput. Graph. Appl. 21,). 2001.

[42] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009: Ieee, pp. 248-255.

[43] S. Bozinovski *et al.*, "The influence of pattern similarity and transfer learning upon training of a base perceptron b2," in *Proceedings of Symposium Informatica*, 1976, pp. 3-121.

[44] S. Bozinovski, "Reminder of the first paper on transfer learning in neural networks, 1976," *Informatica*, vol. 44, no. 3, 2020.

[45] Y. G. Kim *et al.*, "Effectiveness of transfer learning for enhancing tumor classification with a convolutional neural network on frozen sections," (in eng), *Sci Rep*, vol. 10, no. 1, p. 21899, 12 2020, doi: 10.1038/s41598-020-78129-0.

[46] H. Le *et al.*, "Utilizing automated breast cancer detection to identify spatial distributions of tumor infiltrating lymphocytes in invasive breast cancer," *The American Journal of Pathology*, 2020.

[47] W. J. YOUDEN, "Index for rating diagnostic tests," (in eng), *Cancer*, vol. 3, no. 1, pp. 32-5, Jan 1950, doi: 10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3.

[48] *Spyder-documentation*. (2009).

[49] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[50] C. R. Harris *et al.*, "Array programming with NumPy," (in eng), *Nature*, vol. 585, no. 7825, pp. 357-362, 09 2020, doi: 10.1038/s41586-020-2649-2.

[51] G. Bradski *et al.*, "OpenCV," *Dr. Dobb's journal of software tools*, vol. 3, 2000.

[52] P. Barrett *et al.*, "matplotlib--A Portable Python Plotting Package," in *Astronomical data analysis software and systems XIV*, 2005, vol. 347, p. 91.

[53] W. McKinney *et al.*, Ed. *Proceedings of the 9th Python in Science Conference, Volume 445, 2010* (Data structures for statistical computing in python). 2010.

[54] S. Xie *et al.*, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492-1500.

[55] M. P. Cohen, "Stratified Sampling," in *International Encyclopedia of Statistical Science*, M. Lovric Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1547-1550.

[56] G. Jiménez *et al.*, "Deep Learning for Semantic Segmentation vs. Classification in Computational Pathology: Application to Mitosis Analysis in Breast Cancer Grading," (in eng), *Front Bioeng Biotechnol,* vol. 7, p. 145, 2019, doi: 10.3389/fbioe.2019.00145.